

Metadata

/*Metadata

Metadata is essentially data about data. Every time you create a database object, the database server needs to record various pieces of information.

Data is collectively known as the data dictionary or system catalog. The database server needs to store this data persistently, and it needs to be able to quickly retrieve this data in order to verify and execute SQL statements.

Data is stored at a special database, such as MySQL's `information_schema` database*/

```
SELECT
    table_name, table_type
FROM
    information_schema.tables
WHERE
    table_schema = 'sakila'
ORDER BY 1;
```

/*Only Base Table*/

```
SELECT
    table_name, table_type
FROM
    information_schema.tables
WHERE
    table_schema = 'sakila'
    AND table_type = 'BASE TABLE'
ORDER BY 1;
```

```
/*Check only views table*/
```

```
SELECT
    table_name, is_updatable
FROM
    information_schema.views
WHERE
    table_schema = 'sakila'
ORDER BY 1;
```

```
SELECT
    *
FROM
    information_schema.views
WHERE
    table_schema = 'sakila'
ORDER BY 1;
```

```
/*Column info of the film table*/
```

```
SELECT
    column_name,
    data_type,
    character_maximum_length AS char_max_len,
    numeric_precision AS num_prcsn,
    numeric_scale AS num_scale
FROM
    information_schema.columns
WHERE
    table_schema = 'sakila'
```

```
        AND table_name = 'film'
ORDER BY ordinal_position;
```

```
/*The ordinal_position column is included merely as a means to
retrieve the columns in the order in which they were added to the
table.*/
```

```
/*information about a table's indexes*/
```

```
SELECT
    index_name, non_unique, seq_in_index, column_name
FROM
    information_schema.statistics
WHERE
    table_schema = 'sakila'
    AND table_name = 'rental'
ORDER BY 1 , 3;
```

```
/*Query that retrieves all of the constraints in the Sakila schema*/
```

```
SELECT
    constraint_name, table_name, constraint_type
FROM
    information_schema.table_constraints
WHERE
    table_schema = 'sakila'
ORDER BY 3 , 1;
```

View name	Provides information about...
schemata	Databases
tables	Tables and views
columns	Columns of tables and views
statistics	Indexes
user_privileges	Who has privileges on which schema objects
schema_privileges	Who has privileges on which databases
table_privileges	Who has privileges on which tables
column_privileges	Who has privileges on which columns of which tables
character_sets	What character sets are available
collations	What collations are available for which character sets
collation_character_set_applicability	Which character sets are available for which collation
table_constraints	The unique, foreign key, and primary key constraints
key_column_usage	The constraints associated with each key column
routines	Stored routines (procedures and functions)
views	Views
triggers	Table triggers
plugins	Server plug-ins
engines	Available storage engines
partitions	Table partitions
events	Scheduled events
processlist	Running processes
referential_constraints	Foreign keys
parameters	Stored procedure and function parameters
profiling	User profiling information

/*Working With Metadata

Deployment Verification

After the deployment scripts have been run, it's a good idea to run a verification script to ensure that the new schema objects are in place with the appropriate columns, indexes, primary keys, and so forth.*/

```
SELECT
    tbl.table_name,
    (SELECT
        COUNT(*)
    FROM
        information_schema.columns clm
    WHERE
        clm.table_schema = tbl.table_schema
        AND clm.table_name = tbl.table_name) num_columns,
    (SELECT
        COUNT(*)
    FROM
        information_schema.statistics sta
    WHERE
        sta.table_schema = tbl.table_schema
        AND sta.table_name = tbl.table_name) num_indexes,
    (SELECT
        COUNT(*)
    FROM
        information_schema.table_constraints tc
    WHERE
        tc.table_schema = tbl.table_schema
        AND tc.table_name = tbl.table_name
```

```

            AND      tc.constraint_type      =      'PRIMARY      KEY')
num_primary_keys
FROM
    information_schema.tables tbl
WHERE
    tbl.table_schema = 'sakila'
    AND tbl.table_type = 'BASE TABLE'
ORDER BY 1;

```

/*Dynamic SQL Generation

MySQL, allow SQL statements to be submitted to the server as strings. Submitting strings to a database engine rather than utilizing its SQL interface is generally known as dynamic SQL execution.*/

```
SET @qry = 'SELECT customer_id, first_name, last_name FROM customer';
```

```
PREPARE dynsql1 FROM @qry;
```

```
EXECUTE dynsql1;
```

```
SET @qry = 'SELECT customer_id, first_name, last_name
```

```
FROM customer WHERE customer_id = ?';
```

```
PREPARE dynsql2 FROM @qry;
```

```
SET @custid = 9;
```

```
EXECUTE dynsql2 USING @custid;
```

```
SET @custid = 145;
```

```
EXECUTE dynsql2 USING @custid;
```

/*Exercise - 1

Write a query that lists all of the indexes in the Sakila schema.
Include the table names.*/

```
SELECT DISTINCT  
    table_name, index_name  
FROM  
    information_schema.statistics  
WHERE  
    table_schema = 'sakila';
```